K 7

```
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRR         RRR  UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  000         000  FFF              FFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  000         000  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  000         000  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  000         000  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
```

```
_$2

Sym
---
NDX
NDX
NUM
NUM
OPE
OUT
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAD
PAG
PAG
PAG
PAG
PAG
PAG
PAG
PER
PUT
RCO
RIN

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

STR
STR
STR
STR
STK
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

```
**FILE**ID**FORMAT                                N 12



          FFFFFFFFFF    000000    RRRRRRRR    MM      MM    AAAAAA    TTTTTTTTTT
          FFFFFFFFFF    000000    RRRRRRRR    MM      MM    AAAAAA    TTTTTTTTTT
          FF          00      00  RR      RR  MMMM  MMMM  AA      AA      TT
          FF          00      00  RR      RR  MMMM  MMMM  AA      AA      TT
          FF          00      00  RR      RR  MM  MM  MM  AA      AA      TT
          FF          00      00  RR      RR  MM  MM  MM  AA      AA      TT
          FFFFFFFF    00      00  RRRRRRRR    MM      MM  AA      AA      TT
          FFFFFFFF    00      00  RRRRRRRR    MM      MM  AA      AA      TT
          FF          00      00  RR  RR      MM      MM  AAAAAAAAAA      TT
          FF          00      00  RR  RR      MM      MM  AAAAAAAAAA      TT
          FF          00      00  RR    RR    MM      MM  AA      AA      TT          ....
          FF          00      00  RR    RR    MM      MM  AA      AA      TT          ....
          FF            000000    RR      RR  MM      MM  AA      AA      TT          ....
          FF            000000    RR      RR  MM      MM  AA      AA      TT          ....


          LL            IIIIII      SSSSSSSS
          LL            IIIIII      SSSSSSSS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II        SSSSSS
          LL              II        SSSSSS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LLLLLLLLLL    IIIIII    SSSSSSSS
          LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %TITLE 'FORMAT - generate formatted output lines'
0 !<BLF/NOFORMAT>
0
0 MODULE format (IDENT = 'V04-000'
0                 %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE)]
0                 ) =
1 BEGIN
1
1 !<BLF/FORMAT>
1 !<BLF/LOWERCASE_USER>
1 !<BLF/UPPERCASE_KEY>
1 !<BLF/MACRO>
1 !
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1
1 !++
1 ! FACILITY:
1 !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRTOC/CONTENTS Utility
1 !
1 ! ABSTRACT:
1 !   Generate formatted output lines
1 !
1 ! ENVIRONMENT:   Transportable
1 !
1 ! AUTHOR:        JPK
1 !
1 ! CREATION DATE: March 1982
1 !
1 ! MODIFIED BY:
1 !
1 !       005     JPK00008        09-Mar-1983
1 !               Modified CONTENTS and CAPTION to support new BRN formats,
1 !               support SEND CONTENTS, /DOUBLE_SPACE, page numbered chapters,
1 !               guarantee space after section number and to write new
```

```
  1 !                   prologue and epilog for RUNOFF output.
  1 !                   Modified FORMAT to quote only the RUNOFF flags used by CONTENTS.
  1 !                   Modified CNTVMS to fix default for /DOUBLE_SPACE and do more
  1 !                   value checking.
  1 !
  1 !       004         JPK00007        14-Feb-1983
  1 !                   Global edit of all sources for CONTENTS/DSRTOC:
  1 !                   - module names are now consistant with file names
  1 !                   - copyright dates have been updated
  1 !                   - facility names have been updated
  1 !                   - revision history was updated to be consistant with DSR/DSRPLUS
  1 !
  1 !       003         JPK00006        14-Feb-1983
  1 !                   Modified CNTVMS, CONTENTS, FORMAT and CNTVMSMSG to generate
  1 !                   error messages for DSRTOC or CONTENTS depending on the
  1 !                   compiletime variant for DSRPLUS (/VARIANT:8192)
  1 !
  1 !       002         JPK00004        11-Feb-1983
  1 !                   Changed the global variable name INDENT to LINE_INDENT in
  1 !                   modules CONTENTS, CAPTION, FORMAT and GBLDCL.
  1 !                   Removed declarations of PDENTS in modules CNTVMS, CONTENTS,
  1 !                   and CAPTION and replaced with a module wide BIND using the
  1 !                   new name INDENTS.
  1 !                   Changed handling of INDENTS [1]. It no longer represents the
  1 !                   sum of the chapter and title indents.
  1 !
  1 !--
  1
  1 !
  1 ! TABLE OF CONTENTS:
  1 !
  1
  1 FORWARD ROUTINE
  1     insref : NOVALUE,                               ! Insert a page reference
  1     fmttxt : NOVALUE,                               ! Format and output text
  1     endwrd : NOVALUE,                               ! Verify word fits on line
  1     split : NOVALUE;                                ! Start new output file for TMS
  1
  1 !
  1 ! INCLUDE FILES:
  1 !
  1
  1 LIBRARY 'NXPORT:XPORT';
  1
L 1 %IF %BLISS (BLISS32)
  1 %THEN
  1
  1 REQUIRE 'REQ:CNTVMSREQ';
  1 %FI
  1
  1 REQUIRE 'REQ:TOCRTY';                             ! Table of Contents file formats
  1
  1 REQUIRE 'REQ:CNTCLI';                             ! Command line information block formats
  1
  1
  1 !
  1 ! MACROS:
  1 !
  1
  1 MACRO
  1     !
  1     ! Write a character to output line
  1     !
M 1     write_char (ch) [] =
M 1         BEGIN
M 1         CH$WCHAR_A (ch, lp);
M 1         intlin = .intlin + 1;
M 1
M 1         %IF NOT %NULL (%REMAINING)
M 1         %THEN
M 1             extlin = .extlin + 1;
M 1         %FI
M 1
M 1         END
  1     %,
  1     !
  1     ! Write a text literal to the output line
  1     !
M 1     literal_text (str) =
M 1         BEGIN
M 1         CH$MOVE (%CHARCOUNT (str), CH$PTR (UPLIT (str)), .lp);
M 1         lp = CH$PLUS (.lp, %CHARCOUNT (str));
M 1         intlin = .intlin + %CHARCOUNT (str);
M 1         END
  1     %,
  1     !
  1     ! Pad the output line with blanks
  1     !
M 1     pad (n_blanks) =
M 1         BEGIN
M 1
M 1         IF n_blanks GTR 0
M 1         THEN
M 1             BEGIN
M 1             CH$FILL (%C' ', n_blanks, .lp);
M 1             lp = CH$PLUS (.lp, n_blanks);
M 1             intlin = .intlin + n_blanks;
M 1             extlin = .extlin + n_blanks;
M 1             END;
M 1
M 1         END
  1     %,
  1     !
  1     ! Clear the text lines being built up.
  1     !
M 1     clr_line (_) =
M 1         BEGIN
M 1         lp = CH$PTR (line);
M 1         intlin = 0;
M 1         extlin = 0;
M 1         END
  1     %,
  1     !
```

```
	! Insert specified character sequence into file, as is.
	!
	put (str) =
	    BEGIN
	    $str_copy (string = str, target = tmpstr);
	    chrout = .chrout + .tmpstr [str$h_length];
	    $xpo_put (iob = tocoob, string = tmpstr);
	    !
	    ! For TMS output, split the output file if it gets too large
	    !

	    IF .cmdblk [contents$v_tms11] THEN split ();

	    END
	%;

!
! EQUATED SYMBOLS:
!

LITERAL
	tms_characters_per_file = 20*512,		! TMS files may be 20 blocks long
	rintes = %O'34' : UNSIGNED (8),
	true = 1,
	false = 0;

!
! OWN STORAGE:
!

OWN						! Storage for remembering words.
	fileno : INITIAL (0),			! Output file number
	outfile : $str_descriptor (class = dynamic, string = (0,0)), ! Save output filename here
	wrdptr,					! CH$PTR to start of current word.
	extwrd,					! Number of print positions in current word.
	intwrd;					! Number of characters needed to represent current word.

!
! EXTERNAL REFERENCES:
!

EXTERNAL
	cmdblk : $contents_cmd,			! Command line information block
	tocoob : $xpo_iob (),			! IOB for the resulting .RNT file
	chrout,					! Number of characters written to output file
	tmpstr : $str_descriptor (),		! For temporary strings
	hl_n,					! "n" from latest .HL n command
	major,					! Major record type code
	lp,					! CH$PTR along line being built up
	intlin,					! Number of characters needed to represent text
	extlin,					! Number of resulting print positions
	line : VECTOR [CH$ALLOCATION (10000)],	! Buffer in which line is being built up.
	lenpag,					! Number of characters in the converted page number.
	txtpag : VECTOR [CH$ALLOCATION (50)],	! The text (lots of room)
	rmargin,				! Used by ENDWRD for controlling filling lines.
	wrap,					! Wrap long lines around to here.
	line_indent;				! Assume this standard indentation before the text.
```

```
;   229          0704  1
;   230        L 0705  1 %IF %BLISS (BLISS32)
;   231          0706  1 %THEN
;   232          0707  1
;   233          0708  1 EXTERNAL ROUTINE
;   234          0709  1     open_error;                                  ! File open error handler
;   235          0710  1
;   236          0711  1 %FI
```

```
1 %SBTTL 'INSREF - insert page reference into line'
1 GLOBAL ROUTINE insref : NOVALUE =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine inserts a page reference into the output line
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       None
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       cmdblk  - command line information block
1 !       hl_n    - current header level
1 !       rmargin - right margin
1 !       extlin  - external line length
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       wrap    - line wrap point
1 !       rmargin - right margin
1 !               - variables related to output line
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
1
2     BEGIN
2
2     IF .hl_n GTR .cmdblk [contents$g_page_level] THEN RETURN;
2
2     IF .cmdblk [contents$v_tms11]
2     THEN
3         write_char (%C'@')
2     ELSE
3         BEGIN
3         !
3         ! OK. User wants this header to show dots and page number.
3         ! Insert a sequence of alternating dots and spaces out to where the
3         ! page number will go.
3         ! First force a space to follow the last text character.
3         !
3
3         IF .extlin LSS .rmargin
3         THEN
4             BEGIN
4             !
4             ! The position of the last character of the text
4             ! is not inside where the page number goes.
```

```
  4                   !
  4                   write_char (%C' ', counts_visually);
  3                   END;
  3
  3               !
  3               ! Now start inserting the dot-space sequence
  3               !
  3
  3               INCR i FROM (.extlin + 1) TO .rmargin DO
  3                   !
  3                   ! Insert a space for odd columns, a dot for even ones.
  3                   !
P 3                   write_char ((IF .i THEN %C' ' ELSE CH$RCHAR (CH$PTR (cmdblk [contents$c_leader_char]))),
  3                       counts_visually);
  3
  3               !
  3               ! Insert a space following the sequence so there's
  3               ! no dot just before the page number.
  3               !
  3               write_char (%C' ', counts_visually);
  3               !
  3               ! Before putting the page number through the standard text processor,
  3               ! set the wrap point in case the page number doesn't fit.
  3               !
  3               wrap = .cmdblk [contents$g_page_width] - .lenpag;
  3               !
  3               ! Allow the page number to cause the line to be
  3               ! filled out to the full page width.
  3               !
  3               rmargin = .cmdblk [contents$g_page_width];
  2               END;
  2
  2           !
  2           ! Finally, insert the generated page number onto the line
  2           !
  2           fmttxt (.lenpag, CH$PTR (txtpag));
  2
  2           IF .cmdblk [contents$v_tms11] THEN write_char (%C'@');
  2
  1           END;


                                                   .TITLE  FORMAT FORMAT - generate formatted output lines
                                                   .IDENT  \V04-000\

                                                   .PSECT  $OWN$,NOEXE,2

                   00000000  00000 FILENO:  .LONG   0                                        :
                       0000  00004 OUTFILE:.WORD    0                                        :
                      02 0E  00006          .BYTE   14, 2                                    :
                   00000000  00008          .LONG   0                                        :
                             0000C WRDPTR:  .BLKB   4
                             00010 EXTWRD:  .BLKB   4
                             00014 INTWRD:  .BLKB   4

                                            .EXTRN  DSRTOC$_BADVALUE
                                            .EXTRN  DSRTOC$_OPENIN, DSRTOC$_OPENOUT
```

```
                                                                    .EXTRN  DSRTOC$_VALERR, DSRTOC$_CAPTIONS
                                                                    .EXTRN  DSRTOC$_CLOSEQUOT
                                                                    .EXTRN  DSRTOC$_CONFQUAL
                                                                    .EXTRN  DSRTOC$_CTRLCHAR
                                                                    .EXTRN  DSRTOC$_EMPTYIN
                                                                    .EXTRN  DSRTOC$_IGNORED
                                                                    .EXTRN  DSRTOC$_INVINPUT
                                                                    .EXTRN  DSRTOC$_INVRECORD
                                                                    .EXTRN  DSRTOC$_OVERSTRK
                                                                    .EXTRN  DSRTOC$_COMPLETE
                                                                    .EXTRN  DSRTOC$_CREATED
                                                                    .EXTRN  DSRTOC$_IGNORENEW
                                                                    .EXTRN  DSRTOC$_IGNOREOLD
                                                                    .EXTRN  DSRTOC$_PROCFILE
                                                                    .EXTRN  DSRTOC$_TEXTD, DSRTOC$_TMS11
                                                                    .EXTRN  DSRTOC$_IDENT, CMDBLK
                                                                    .EXTRN  TOCOOB, CHROUT, TMPSTR
                                                                    .EXTRN  HL_N, MAJOR, LP
                                                                    .EXTRN  INTLIN, EXTLIN, LINE
                                                                    .EXTRN  LENPAG, TXTPAG, RMARGIN
                                                                    .EXTRN  WRAP, LINE_INDENT
                                                                    .EXTRN  OPEN_ERROR

                                                                    .PSECT  $CODE$,NOWRT,2

                                                   00FC 00000       .ENTRY  INSREF, Save R2,R3,R4,R5,R6,R7                   ; 0713
                          57 00000000G      EF   9E 00002           MOVAB   LENPAG, R7                                       ;
                          56 00000000G      EF   9E 00009           MOVAB   RMARGIN, R6                                      ;
                          55 00000000G      EF   9E 00010           MOVAB   INTLIN, R5                                       ;
                          54 00000000G      EF   9E 00017           MOVAB   EXTLIN, R4                                       ;
                          53 00000000G      EF   9E 0001E           MOVAB   CMDBLK, R3                                       ;
                          52 00000000G      EF   9E 00025           MOVAB   LP, R2                                           ;
                     08   A3 00000000G      EF   D1 0002C           CMPL    HL_N, CMDBLK+8                                   ; 0749
                                            6C   14 00034           BGTR    8$                                               ;
               07         63                01   E1 00036           BBC     #1, CMDBLK, 1$                                   ; 0751
                     00   B2          40    8F   90 0003A           MOVB    #64, @LP                                         ; 0753
                                            41   11 0003F           BRB     7$                                               ; 0751
                          66                64   D1 00041 1$:       CMPL    EXTLIN, RMARGIN                                  ; 0763
                                            0A   18 00044           BGEQ    2$                                               ;
                     00   B2                20   90 00046           MOVB    #32, @LP                                         ; 0770
                                            62   D6 0004A           INCL    LP                                               ;
                                            65   D6 0004C           INCL    INTLIN                                           ;
                                            64   D6 0004E           INCL    EXTLIN                                           ;
                          51                64   D0 00050 2$:       MOVL    EXTLIN, I                                        ; 0777
                                            16   11 00053           BRB     6$                                               ;
                          05                51   E9 00055 3$:       BLBC    I, 4$                                            ; 0782
                          50                20   D0 00058           MOVL    #32, R0                                          ;
                                            04   11 0005B           BRB     5$                                               ;
                          50          02    A3   9A 0005D 4$:       MOVZBL  CMDBLK+2, R0                                     ;
                     00   B2                50   90 00061 5$:       MOVB    R0, @LP                                          ;
                                            62   D6 00065           INCL    LP                                               ;
                                            65   D6 00067           INCL    INTLIN                                           ;
                                            64   D6 00069           INCL    EXTLIN                                           ;
               E6         51                66   F3 0006B 6$:       AOBLEQ  RMARGIN, I, 3$                                   ; 0777
                     00   B2                20   90 0006F           MOVB    #32, @LP                                         ; 0788
                                            64   D6 00073           INCL    EXTLIN                                           ;
   00000000G   EF    18   A3                67   C3 00075           SUBL3   LENPAG, CMDBLK+24, WRAP                          ; 0793
```

```
                               66         18   A3  D0 0007E        MOVL    CMDBLK+24, RMARGIN                          : 0798
                                               62  D6 00082 7$:    INCL    LP                                          : 0753
                                               65  D6 00084        INCL    INTLIN                                      :
                                   00000000G   EF  9F 00086        PUSHAB  TXTPAG                                      : 0804
                                               67  DD 0008C        PUSHL   LENPAG                                      :
                  00000000V   EF               02  FB 0008E        CALLS   #2, FMTTXT                                  :
            09                63               01  E1 00095        BBC     #1, CMDBLK, 8$                              : 0806
                         00   B2          40   8F  90 00099        MOVB    #64, @LP                                    :
                                               62  D6 0009E        INCL    LP                                          :
                                               65  D6 000A0        INCL    INTLIN                                      :
                                                   04 000A2 8$:    RET                                                 : 0808

; Routine Size:  163 bytes,    Routine Base:  $CODE$ + 0000
```

```
%SBTTL 'FMTTXT - scan and format text'
GLOBAL ROUTINE fmttxt (txt_len, txt_ptr) : NOVALUE =
!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine scans the input text and formats it into line.
!
!       Special characters are quoted for RUNOFF output or changed to the
!       appropriate sequence for TMS unless the text is from a .SEND TOC.
!
!       Special characters from .SEND TOC are inserted as is for RUNOFF.
!
!       Special characters from .SEND TOC are inserted as is for TMS
!       with the exception of '>' which is inserted as '>', the line
!       is broken, and a new line is started with '<'. This is because
!       .SEND TOC text is inserted as a comment for TMS.
!
!       Emphasis in the input string is kept if emphasis is enabled
!       for the current header level or if the text is from a .SEND TOC.
!
! FORMAL PARAMETERS:
!
!       txt_len         - length of input string
!       txt_ptr         - CH$PTR to input string
!
! IMPLICIT INPUTS:
!
!       cmdblk          - command line information block
!       major           - type of text being processed
!       hl_n            - header level number being processed
!       lp              - CH$PTR to next character position in line
!       intlin          - internal line length
!       extlin          - external line length
!       rmargin         - indicates how far to the right text may be inserted
!       wrap            - column to wrap a broken line to
!       line_indent     - number of columns which line is indented
!
! IMPLICIT OUTPUTS:
!
!       lp              - points to next available character position in line
!       intlin          - reflects new internal length
!       extlin          - reflects new external length
!       wrdptr          - set to initial value of lp
!       intwrd          - set to initial value of intlin
!       extwrd          - set to initial value of extlin
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       None
!
! SIDE EFFECTS:
!
!       None
!
!--
```

```
BEGIN

LOCAL
    keep_bold,
    keep_und,
    doing_bold,
    doing_und,
    bold_char,
    und_char,
    open_quote,
    ptr,
    len;

!
! Keep bolding if .SEND TOC and user said /BOLD=anything
! or if the hl value of the text LEQ the user specified level.
!
IF ((.major EQL maj_send) AND (.cmdblk [contents$g_bold] NEQ -1))
    OR (.hl_n LEQ .cmdblk [contents$g_bold])
THEN
    keep_bold = true
ELSE
    keep_bold = false;

!
! Keep underlining if .SEND TOC and user said /UNDERLINE=anything
! or if the hl value of the text LEQ the user specified level.
!
IF ((.major EQL maj_send) AND (.cmdblk [contents$g_underline] NEQ -1))
    OR (.hl_n LEQ .cmdblk [contents$g_underline])
THEN
    keep_und = true
ELSE
    keep_und = false;

len = .txt_len;                                 ! Copy string length
ptr = .txt_ptr;                                 ! and pointer
wrdptr = .tp;                                   ! Initialize word pointer
intwrd = .intlin;                               ! internal word length
extwrd = .extlin;                               ! external word length
doing_bold = false;                             ! Bold is off
doing_und = false;                              ! as is underlining
bold_char = false;                              ! Character is not bold
und_char = false;                               ! or underlined
open_quote = true;                              ! First '"' we see is an open quote

WHILE .len GTR 0 DO                             ! Process whole input string
    BEGIN

    LOCAL
        ch;

    ch = CH$RCHAR_A (ptr);                      ! Get next character
    len = .len - 1;                             ! one less character

    IF .ch EQL rintes
    THEN
```

```
  4             BEGIN
  4             !
  4             ! RUNOFF internal escape sequence
  4             !
  4
  4             LOCAL
  4                 fnc,
  4                 op;
  4
  4             fnc = CH$RCHAR_A (ptr);                 ! Get function
  4             op = CH$RCHAR_A (ptr);                  ! and operand
  4             len = .len - 2;                         ! 2 less characters to process
  4
  4             SELECTONE .fnc OF
  4                 SET
  4
  4                 [%C'O'] :
  5                     BEGIN
  5                     !
  5                     ! Overstrike
  5                     !
  5                     write_char (.op, counts_visually);
  5
  5                     IF .cmdblk [contents$v_tms11]
  5                     THEN
  6                         BEGIN
  6                         !
  6                         ! Overstriking is frowned upon for TMS
  6                         !
L 6 %IF %BLISS (BLISS32)
  6 %THEN
  6                         SIGNAL (contents$_overstrk, 0, contents$_textd, 2, .txt_len, .txt_ptr);
U 6 %ELSE
U 6                         $xpo_put_msg (severity = warning,
U 6                             string = 'the following line contains an overstrike sequence',
U 6                             string = (.txt_len, .txt_ptr));
  6 %FI
  6
  6                         literal_text ('[ec]');
  6                         END
  5                     ELSE
  5                         write_char (%C'%');
  5
  4                     END;
  4
  4                 [%C'B'] :
  4                     !
  4                     ! Bold next character if keeping bold
  4                     !
  4                     bold_char = (IF .keep_bold THEN true ELSE false);
  4
  4                 [%C'U'] :
  4                     !
  4                     ! Underline next character if keeping underlining
  4                     !
  4                     und_char = (IF .keep_und THEN true ELSE false);
  4
```

```
4                       [OTHERWISE] :
4                           !
4                           ! Unknown sequence - do nothing
4                           !
4                           ;
4                       TES;
4
4                   END
3               ELSE
4                   BEGIN
4                   !
4                   ! A 'normal' character
4                   !
4
4                   IF NOT .bold_char
4                   THEN
5                       BEGIN
5                       !
5                       ! Do not bold this character
5                       !
5
6                       IF .doing_bold AND (.ch NEQ %C' ')
5                       THEN
6                           BEGIN
6                           !
6                           ! Bold is turned on and the current character is non-blank
6                           ! Turn off bold
6                           !
6
6                           IF .cmdblk [contents$v_tms11] THEN literal_text ('[fr') ELSE literal_text ('\*');
6
6                           IF .doing_und
6                           THEN
7                               BEGIN
7                               !
7                               ! Must turn underlining off too on since both bold
7                               ! and underline use the same termination sequence
7                               !
7
7                               IF .cmdblk [contents$v_tms11] THEN literal_text ('fr') ELSE literal_text ('\&');
7
7                               IF .und_char
7                               THEN
8                                   BEGIN
8                                   !
8                                   ! This character is underlined
8                                   ! Turn underlining back on.
8                                   !
8
8                                   IF .cmdblk [contents$v_tms11] THEN literal_text ('fi') ELSE literal_text ('^&');
8
8                                   END
7                               ELSE
7                                   !
7                                   ! Character is not underlined
7                                   ! Note that we've turned off underlining
7                                   !
```

```
                doing_und = false;

            END;

        IF .cmdblk [contents$v_tms11] THEN write_char (%C']');

        doing_bold = false;
        END;

    END
ELSE
    BEGIN
    !
    ! Bold next character
    !

    IF NOT .doing_bold
    THEN
        BEGIN
        !
        ! Turn on bolding
        !

        IF .cmdblk [contents$v_tms11] THEN literal_text ('[fb]') ELSE literal_text ('^*');

        doing_bold = true;
        END;

    bold_char = false;              ! Reset bold character flag
    END;

IF NOT .und_char
THEN
    BEGIN
    !
    ! Do not underline this character
    !

    IF .doing_und AND (.ch NEQ %C' ')
    THEN
        BEGIN
        !
        ! Underlining is turned on and the current character is
        ! non-blank. Turn off underlining.
        !

        IF .cmdblk [contents$v_tms11] THEN literal_text ('[fr') ELSE literal_text ('\&');

        IF .cmdblk [contents$v_tms11]
        THEN
            BEGIN
            !
            ! If bolding is on, turn it off and back on since
            ! both bold and underline user the same terminators
            !

            IF .doing_bold THEN literal_text ('frfb');
```

```
7
7                           write_char (%C']');
6                           END;
6
6                   doing_und = false;
5                   END;
5
5           END
4       ELSE
5           BEGIN
5           !
5           ! Underline next character
5           !
5
5           IF NOT .doing_und
5           THEN
6               BEGIN
6               !
6               ! Turn on underlining
6               !
6
6               IF .cmdblk [contents$v_tms11] THEN literal_text ('[fi]') ELSE literal_text ('^&');
6
6               doing_und = true;
5               END;
5
5           und_char = false;                ! Reset flag
4           END;
4
4       SELECTONE true OF
4           SET
4
4           [.ch EQL %C' '] :
4               endwrd (true);
4
4           [.ch EQL %C'>'] :
5               BEGIN
5
5               IF .major EQL maj_send
5               THEN
6                   BEGIN
6                   write_char (.ch, counts_visually);
6
6                   IF .cmdblk [contents$v_tms11]
6                   THEN
7                       BEGIN
7                       !
7                       ! For TMS, a '>' in a SEND TOC starts a new line
7                       !
7                       put ((.intlin, CH$PTR (line)));
7                       clr_line ();
7                       write_char (%C'<');
7                       wrdptr = .lp;
7                       extwrd = .extlin;
7                       intwrd = .intlin;
6                       END;
6
```

```
                                    END
                                ELSE
                                    BEGIN

                                    IF .cmdblk [contents$v_tms11]
                                    THEN
                                        literal_text ('+z')
                                    ELSE
                                        write_char (.ch, counts_visually);

                                    END;

                                END;

                            [(.ch LSS %C' ') OR (.ch GTR %O'176')] :
                                BEGIN
                                !
                                ! A control character
                                !

                                IF .cmdblk [contents$v_tms11]
                                THEN
                                    BEGIN
                                    !
                                    ! Control characters are ignored for TMS output
                                    !
%IF %BLISS (BLISS32)
%THEN
                                    SIGNAL (contents$_ctrlchar, 0, contents$_textd, 2, .txt_len, .txt_ptr);
%ELSE
                                    $xpo_put_msg (severity = warning,
                                        string = 'the following line contains control characters which were ignored',
                                        string = (.txt_len, .txt_ptr));
%FI

                                    END
                                ELSE
                                    BEGIN
                                    !
                                    ! For RUNOFF output
                                    !
                                    write_char (%C'_');         ! Quote the character
                                    write_char (.ch);           ! Write the character itself

                                    IF .ch EQL %O'10'
                                    THEN                        ! Backspace shortens the external length
                                        extlin = .extlin - 1;

                                    END;

                                END;

                            [OTHERWISE] :
                                BEGIN
                                !
                                ! For every thing else...
                                !
```

```
5
5                       IF .major EQL maj_send
5                       THEN
6                           BEGIN
6                           !
6                           ! Just write the character for .SEND TOC
6                           !
6                           write_char (.ch, counts_visually);
6                           END
5                       ELSE
6                           BEGIN
6                           !
6                           ! Check for special characters
6                           !
6                           IF .cmdblk [contents$v_tms11]
6                           THEN
7                               BEGIN
7                               !
7                               ! For TMS...
7                               !
7                               SELECTONE .ch OF
7                                   SET
7
7                                   [%C' '] :
7                                       literal_text ('*n10*');
7
7                                   [%C'-'] :
7                                       literal_text ('+n');
7
7                                   [%C'*'] :
7                                       literal_text ('+a');
7
7                                   [%C'='] :
7                                       literal_text ('+e');
7
7                                   [%C'+'] :
7                                       literal_text ('+p');
7
7                                   [%C'\'] :
7                                       literal_text ('+s');
7
7                                   [%C'@'] :
7                                       literal_text ('+t');
7
7                                   [%C'/'] :
7                                       literal_text ('+.');
7
7                                   [%C'|'] :
7                                       literal_text ('+v');
7
7                                   [%C'{'] :
7                                       literal_text ('+w');
7
7                                   [%C'}'] :
7                                       literal_text ('+x');
```

```
                        [%C'<'] :
                            literal_text ('+y');

                        [%C'['] :
                            literal_text ('+(');

                        [%C']'] :
                            literal_text ('+)');

                        [%C'"'] :
                            BEGIN

                            IF .open_quote
                            THEN
                                BEGIN
                                !
                                ! Opening quote of quoted string
                                !
                                literal_text ('"');
                                open_quote = false;     ! Next quote is not an open quote
                                END
                            ELSE
                                BEGIN
                                !
                                ! Closing quote
                                !
                                literal_text ('''''');
                                open_quote = true;      ! Next quote is open quote
                                END;

                            END;

                        [OTHERWISE] :
                            !
                            ! A real normal character
                            !
                            write_char (.ch, counts_visually);
                        TES;

                    END
                ELSE
                    BEGIN
                    !
                    ! For RUNOFF
                    !

                    IF (.ch EQL %C'_')          ! ACCEPT flag
                        OR (.ch EQL %C'*')      ! BOLD flag
                        OR (.ch EQL %C'!')      ! COMMENT flag
                        OR (.ch EQL %C'.')      ! CONTROL flag
                        OR (.ch EQL %C'\')      ! LOWERCASE flag
                        OR (.ch EQL %C'%')      ! OVERSTRIKE flag
                        OR (.ch EQL %C'&')      ! UNDERLINE flag
                        OR (.ch EQL %C'^')      ! UPPERCASE flag
                    THEN
                        !
```

```
  7                 ! A RUNOFF flag. Quote it.
  7                 !
  7                 write_char (%C'_');
  7
  7               write_char (.ch, counts_visually);
  6               END;
  6
  5             END;
  5
  4           END;
  4         TES;
  4
  3       END;
  3
  2     END;
  2
  2   endwrd (false);                         ! Check to see if word fits
  2
  2   IF .cmdblk [contents$v_tms11]
  2   THEN
  3     BEGIN
  3
  3     IF .doing_bold OR .doing_und
  3     THEN
  4       BEGIN
  4       !
  4       ! Turn off one of them.
  4       !
  4       literal_text ('[fr');
  4       !
  4       ! If doing both bold and underline, turn off the other.
  4       !
  4
  4       IF .doing_bold AND .doing_und THEN literal_text ('fr');
  4
  4       write_char (%C']');
  3       END;
  3
  3     IF NOT .open_quote
  3     THEN
  3       !
  3       ! Missing a close quote
  3       !
L 3 %IF %BLISS (BLISS32)
  3 %THEN
  3       SIGNAL (contents$_closequot, 0, contents$_textd, 2, .txt_len, .txt_ptr);
U 3 %ELSE
U 3       $xpo_put_msg (severity = warning,
U 3         string = 'the following text is missing a close quote',
U 3         string = (.txt_len, .txt_ptr));
  3 %FI
  3
  3     END
  2   ELSE
  3     BEGIN
  3     !
  3     ! For RUNOFF
```

```
;   906        1379   3             !
;   907        1380   3
;   908        1381   3             IF .doing_bold THEN literal_text ('\*'); ! Turn off bolding
;   909        1382   3
;   910        1383   3             IF .doing_und THEN literal_text ('\&'); ! Turn off underlining
;   911        1384   3
;   912        1385   2             END;
;   913        1386   2
;   914        1387   1         END;



                                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

                            5D  63  65  5B  00000 P.AAA:  .ASCII  \[ec]\                                  ;
                            00  72  66  5B  00004 P.AAB:  .ASCII  \[fr\<0>                                ;
                            00  00  2A  5C  00008 P.AAC:  .ASCII  <92>\*\<0><0>                           ;
                            00  00  72  66  0000C P.AAD:  .ASCII  \fr\<0><0>                              ;
                            00  00  26  5C  00010 P.AAE:  .ASCII  <92>\&\<0><0>                           ;
                            00  00  69  66  00014 P.AAF:  .ASCII  \fi\<0><0>                              ;
                            00  00  26  5E  00018 P.AAG:  .ASCII  \^&\<0><0>                              ;
                            5D  62  66  5B  0001C P.AAH:  .ASCII  \[fb]\                                  ;
                            00  00  2A  5E  00020 P.AAI:  .ASCII  \^*\<0><0>                              ;
                            00  72  66  5B  00024 P.AAJ:  .ASCII  \[fr\<0>                                ;
                            00  00  26  5C  00028 P.AAK:  .ASCII  <92>\&\<0><0>                           ;
                            62  66  72  66  0002C P.AAL:  .ASCII  \frfb\                                  ;
                            5D  69  66  5B  00030 P.AAM:  .ASCII  \[fi]\                                  ;
                            00  00  26  5E  00034 P.AAN:  .ASCII  \^&\<0><0>                              ;
                            00  00  7A  2B  00038 P.AAO:  .ASCII  \+z\<0><0>                              ;
            00  00  00  2A  30  31  6E  2A  0003C P.AAP:  .ASCII  \*n10*\<0><0><0>                        ;
                            00  00  6E  2B  00044 P.AAQ:  .ASCII  \+n\<0><0>                              ;
                            00  00  61  2B  00048 P.AAR:  .ASCII  \+a\<0><0>                              ;
                            00  00  65  2B  0004C P.AAS:  .ASCII  \+e\<0><0>                              ;
                            00  00  70  2B  00050 P.AAT:  .ASCII  \+p\<0><0>                              ;
                            00  00  73  2B  00054 P.AAU:  .ASCII  \+s\<0><0>                              ;
                            00  00  74  2B  00058 P.AAV:  .ASCII  \+t\<0><0>                              ;
                            00  00  2E  2B  0005C P.AAW:  .ASCII  \+.\<0><0>                              ;
                            00  00  76  2B  00060 P.AAX:  .ASCII  \+v\<0><0>                              ;
                            00  00  77  2B  00064 P.AAY:  .ASCII  \+w\<0><0>                              ;
                            00  00  78  2B  00068 P.AAZ:  .ASCII  \+x\<0><0>                              ;
                            00  00  79  2B  0006C P.ABA:  .ASCII  \+y\<0><0>                              ;
                            00  00  28  2B  00070 P.ABB:  .ASCII  \+(\<0><0>                              ;
                            00  00  29  2B  00074 P.ABC:  .ASCII  \+)\<0><0>                              ;
                            00  00  22  22  00078 P.ABD:  .ASCII  \""\<0><0>                              ;
                            00  00  27  27  0007C P.ABE:  .ASCII  \''\<0><0>                              ;
                            00  72  66  5B  00080 P.ABF:  .ASCII  \[fr\<0>                                ;
                            00  00  72  66  00084 P.ABG:  .ASCII  \fr\<0><0>                              ;
                            00  00  2A  5C  00088 P.ABH:  .ASCII  <92>\*\<0><0>                           ;
                            00  00  26  5C  0008C P.ABI:  .ASCII  <92>\&\<0><0>                           ;

                                                          .EXTRN  XST$COPY, STR$FAILURE
                                                          .EXTRN  XPO$PUT, XPO$FAILURE

                                                          .PSECT  $CODE$,NOWRT,2

                                           0FFC 00000     .ENTRY  FMTTXT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-   ; 0810
                                                                  R11                                          ;
```

```
                     5E                  18  C2 00002          SUBL2   #24, SP                          ;
                                         50  D4 00005          CLRL    R0                               ; 0883
                     03 00000000G        EF  D1 00007          CMPL    MAJOR, #3                        ;
                                         0F  12 0000E          BNEQ    1$                               ;
                                         50  D6 00010          INCL    R0                               ;
            FFFFFFFF 8F 00000000G        EF  D1 00012          CMPL    CMDBLK+16, #-1                   ;
                                         0D  12 0001D          BNEQ    2$                               ;
           00000000G EF 00000000G        EF  D1 0001F 1$:      CMPL    HL_N, CMDBLK+16                  ; 0884
                                         05  14 0002A          BGTR    3$                               ;
                     6E                  01  D0 0002C 2$:      MOVL    #1, KEEP_BOLD                    ; 0886
                                         02  11 0002F          BRB     4$                               ;
                                         6E  D4 00031 3$:      CLRL    KEEP_BOLD                        ; 0888
                     0D                  50  E9 00033 4$:      BLBC    R0, 5$                           ; 0894
            FFFFFFFF 8F 00000000G        EF  D1 00036          CMPL    CMDBLK+12, #-1                   ;
                                         0D  12 00041          BNEQ    6$                               ;
           00000000G EF 00000000G        EF  D1 00043 5$:      CMPL    HL_N, CMDBLK+12                  ; 0895
                                         06  14 0004E          BGTR    7$                               ;
                  0C AE                  01  D0 00050 6$:      MOVL    #1, KEEP_UND                     ; 0897
                                         03  11 00054          BRB     8$                               ;
                                 0C      AE  D4 00056 7$:      CLRL    KEEP_UND                         ; 0899
                     5B          04      AC  D0 00059 8$:      MOVL    TXT_LEN, LEN                     ; 0901
                  04 AE          08      AC  D0 0005D          MOVL    TXT_PTR, PTR                     ; 0902
           00000000' EF 00000000G        EF  D0 00062          MOVL    LP, WRDPTR                       ; 0903
           00000000' EF 00000000G        EF  D0 0006D          MOVL    INTLIN, INTWRD                   ; 0904
           00000000' EF 00000000G        EF  D0 00078          MOVL    EXTLIN, EXTWRD                   ; 0905
                                 08      AE  D4 00083          CLRL    DOING_BOLD                       ; 0906
                     59                  01  7D 00086          MOVQ    #1, OPEN_QUOTE                   ; 0910
                                         57  7C 00089          CLRQ    DOING_UND                        ; 0907
                                         5B  D5 0008B 9$:      TSTL    LEN                              ; 0912
                                         03  14 0008D          BGTR    10$                              ;
                                       0612  31 0008F          BRW     91$                              ;
                     56          04      BE  9A 00092 10$:     MOVZBL  @PTR, CH                         ; 0918
                                 04      AE  D6 00096          INCL    PTR                              ;
                                         5B  D7 00099          DECL    LEN                              ; 0919
                     1C                  56  D1 0009B          CMPL    CH, #28                          ; 0921
                                         03  13 0009E          BEQL    11$                              ;
                                       00B2  31 000A0          BRW     18$                              ;
                     50          04      BE  9A 000A3 11$:     MOVZBL  @PTR, FNC                        ; 0932
                                 04      AE  D6 000A7          INCL    PTR                              ;
                     51          04      BE  9A 000AA          MOVZBL  @PTR, OP                         ; 0933
                                 04      AE  D6 000AE          INCL    PTR                              ;
                     5B                  02  C2 000B1          SUBL2   #2, LEN                          ; 0934
            0000004F 8F                  50  D1 000B4          CMPL    FNC, #79                         ; 0939
                                         6C  12 000BB          BNEQ    13$                              ;
           00000000G FF                  51  90 000BD          MOVB    OP, @LP                          ; 0944
                        00000000G        EF  D6 000C4          INCL    LP                               ;
                        00000000G        EF  D6 000CA          INCL    INTLIN                           ;
                        00000000G        EF  D6 000D0          INCL    EXTLIN                           ;
        36 00000000G    EF               01  E1 000D6          BBC     #1, CMDBLK, 12$                  ; 0946
                     7E          04      AC  7D 000DE          MOVQ    TXT_LEN, -(SP)                   ; 0954
                                         02  DD 000E2          PUSHL   #2                               ;
                        00000000G        8F  DD 000E4          PUSHL   #DSRTOC$_TEXTD                   ;
                                         7E  D4 000EA          CLRL    -(SP)                            ;
                        00000000G        8F  DD 000EC          PUSHL   #DSRTOC$_OVERSTRK                ;
           00000000G 00                  06  FB 000F2          CALLS   #6, LIB$SIGNAL                   ;
           00000000G FF 00000000'        EF  D0 000F9          MOVL    P.AAA, @LP                       ; 0961
           00000000G EF                  04  C0 00104          ADDL2   #4, LP                           ;
```

```
                                                                  J 14
FORMAT          FORMAT - generate formatted output lines       16-Sep-1984 00:34:26    VAX-11 Bliss-32 V4.0-742      Page  22
V04-000         FMTTXT - scan and format text                  14-Sep-1984 13:06:27    [RUNOFF.SRC]FORMAT.BLI;1            (3)

                 00000000G EF             04 C0 0010B       ADDL2   #4, INTLIN                                 ;
                                          3E 11 00112       BRB     17$                                        ; 0946
                 00000000G FF             25 90 00114 12$:  MOVB    #37, @LP                                   ; 0964
                              00000000G   EF D6 00118       INCL    LP                                         ;
                              00000000G   EF D6 00121       INCL    INTLIN                                     ;
                                          29 11 00127       BRB     17$                                        ; 0936
                 00000042  8F             50 D1 00129 13$:  CMPL    FNC, #66                                   ; 0968
                                          0C 12 00130       BNEQ    15$                                        ;
                           05             6E E9 00132       BLBC    KEEP_BOLD, 14$                             ; 0972
                           5A             01 D0 00135       MOVL    #1, BOLD_CHAR                              ;
                                          18 11 00138       BRB     17$                                        ;
                                          5A D4 0013A 14$:  CLRL    BOLD_CHAR                                  ;
                                          14 11 0013C       BRB     17$                                        ;
                 00000055  8F             50 D1 0013E 15$:  CMPL    FNC, #85                                   ; 0974
                                          0B 12 00145       BNEQ    17$                                        ;
                           05         0C  AE E9 00147       BLBC    KEEP_UND, 16$                              ; 0978
                           58             01 D0 0014B       MOVL    #1, UND_CHAR                               ;
                                          02 11 0014E       BRB     17$                                        ;
                                          58 D4 00150 16$:  CLRL    UND_CHAR                                   ;
                                        FF36 31 00152 17$:  BRW     9$                                         ;
                           03             5A E9 00155 18$:  BLBC    BOLD_CHAR, 19$                             ; 0994
                                        00C3 31 00158       BRW     31$                                        ;
                           03         08  AE E8 0015B 19$:  BLBS    DOING_BOLD, 21$                            ; 1001
                                        0101 31 0015F 20$:  BRW     35$                                        ;
                           20             56 D1 00162 21$:  CMPL    CH, #32                                    ;
                                          F8 13 00165       BEQL    20$                                        ;
                           50 00000000G   EF D0 00167       MOVL    LP, R0                                     ; 1009
51 00000000G  EF           01             01 EF 0016E       EXTZV   #1, #1, CMDBLK, R1                         ;
                           19             51 E9 00177       BLBC    R1, 22$                                    ;
60            18           00 00000000'   EF F0 0017A       INSV    P.AAB, #0, #24, (R0)                       ;
                 00000000G EF             03 C0 00183       ADDL2   #3, LP                                     ;
                 00000000G EF             03 C0 0018A       ADDL2   #3, INTLIN                                 ;
                                          15 11 00191       BRB     23$                                        ;
                           60 00000000'   EF B0 00193 22$:  MOVW    P.AAC, (R0)                                ;
                 00000000G EF             02 C0 0019A       ADDL2   #2, LP                                     ;
                 00000000G EF             02 C0 001A1       ADDL2   #2, INTLIN                                 ;
                           57             57 E9 001A8 23$:  BLBC    DOING_UND, 29$                             ; 1011
                           50 00000000G   EF D0 001AB       MOVL    LP, R0                                     ; 1019
                           09             51 E9 001B2       BLBC    R1, 24$                                    ;
                           60 00000000'   EF B0 001B5       MOVW    P.AAD, (R0)                                ;
                                          07 11 001BC       BRB     25$                                        ;
                           60 00000000'   EF B0 001BE 24$:  MOVW    P.AAE, (R0)                                ;
                 00000000G EF             02 C0 001C5 25$:  ADDL2   #2, LP                                     ;
                 00000000G EF             02 C0 001CC       ADDL2   #2, INTLIN                                 ;
                           2A             58 E9 001D3       BLBC    UND_CHAR, 28$                              ; 1021
                           50 00000000G   EF D0 001D6       MOVL    LP, R0                                     ; 1029
                           09             51 E9 001DD       BLBC    R1, 26$                                    ;
                           60 00000000'   EF B0 001E0       MOVW    P.AAF, (R0)                                ;
                                          07 11 001E7       BRB     27$                                        ;
                           60 00000000'   EF B0 001E9 26$:  MOVW    P.AAG, (R0)                                ;
                 00000000G EF             02 C0 001F0 27$:  ADDL2   #2, LP                                     ;
                 00000000G EF             02 C0 001F7       ADDL2   #2, INTLIN                                 ;
                                          02 11 001FE       BRB     29$                                        ; 1021
                                          57 D4 00200 28$:  CLRL    DOING_UND                                  ; 1037
                           14             51 E9 00202 29$:  BLBC    R1, 30$                                    ; 1041
                 00000000G FF         5D  8F 90 00205       MOVB    #93, @LP                                   ;
                              00000000G   EF D6 0020D       INCL    LP                                         ;
```

```
                          00000000G  EF  D6 00213          INCL    INTLIN                                 ;
                                 08  AE  D4 00219 30$:     CLRL    DOING_BOLD                             ; 1043
                                     45  11 0021C          BRB     35$                                    ; 0994
                      3F         08  AE  E8 0021E 31$:     BLBS    DOING_BOLD, 34$                        ; 1053
                      50  00000000G  EF  D0 00222          MOVL    LP, R0                                 ; 1060
           17 00000000G          EF  01  E1 00229          BBC     #1, CMDBLK, 32$                        ;
                      60  00000000'  EF  D0 00231          MOVL    P.AAH, (R0)                            ;
              00000000G          EF  04  C0 00238          ADDL2   #4, LP                                 ;
              00000000G          EF  04  C0 0023F          ADDL2   #4, INTLIN                             ;
                                     15  11 00246          BRB     33$                                    ;
                      60  00000000'  EF  B0 00248 32$:     MOVW    P.AAI, (R0)                            ;
              00000000G          EF  02  C0 0024F          ADDL2   #2, LP                                 ;
              00000000G          EF  02  C0 00256          ADDL2   #2, INTLIN                             ;
                     08          AE  01  D0 0025D 33$:     MOVL    #1, DOING_BOLD                         ; 1062
                                     5A  D4 00261 34$:     CLRL    BOLD_CHAR                              ; 1065
                      03             58  E9 00263 35$:     BLBC    UND_CHAR, 36$                          ; 1068
                                   0082  31 00266          BRW     42$                                    ;
                      7D             57  E9 00269 36$:     BLBC    DOING_UND, 41$                         ; 1075
                      20             56  D1 0026C          CMPL    CH, #32                                ;
                                     78  13 0026F          BEQL    41$                                    ;
                      50  00000000G  EF  D0 00271          MOVL    LP, R0                                 ; 1083
           19 00000000G          EF  01  E1 00278          BBC     #1, CMDBLK, 37$                        ;
    60     18         00  00000000'  EF  F0 00280          INSV    P.AAJ, #0, #24, (R0)                   ;
              00000000G          EF  03  C0 00289          ADDL2   #3, LP                                 ;
              00000000G          EF  03  C0 00290          ADDL2   #3, INTLIN                             ;
                                     15  11 00297          BRB     38$                                    ;
                      60  00000000'  EF  B0 00299 37$:     MOVW    P.AAK, (R0)                            ;
              00000000G          EF  02  C0 002A0          ADDL2   #2, LP                                 ;
              00000000G          EF  02  C0 002A7          ADDL2   #2, INTLIN                             ;
           31 00000000G          EF  01  E1 002AE 38$:     BBC     #1, CMDBLK, 40$                        ; 1085
                      19         08  AE  E9 002B6          BLBC    DOING_BOLD, 39$                        ; 1093
              00000000G  FF  00000000'  EF  D0 002BA          MOVL    P.AAL, @LP                             ;
              00000000G          EF  04  C0 002C5          ADDL2   #4, LP                                 ;
              00000000G          EF  04  C0 002CC          ADDL2   #4, INTLIN                             ;
              00000000G  FF          5D  8F  90 002D3 39$:     MOVB    #93, @LP                               ; 1095
                          00000000G  EF  D6 002DB          INCL    LP                                     ;
                          00000000G  EF  D6 002E1          INCL    INTLIN                                 ;
                                     57  D4 002E7 40$:     CLRL    DOING_UND                              ; 1098
                                     43  11 002E9 41$:     BRB     46$                                    ; 1068
                      3E             57  E8 002EB 42$:     BLBS    DOING_UND, 45$                         ; 1108
                      50  00000000G  EF  D0 002EE          MOVL    LP, R0                                 ; 1115
           17 00000000G          EF  01  E1 002F5          BBC     #1, CMDBLK, 43$                        ;
                      60  00000000'  EF  D0 002FD          MOVL    P.AAM, (R0)                            ;
              00000000G          EF  04  C0 00304          ADDL2   #4, LP                                 ;
              00000000G          EF  04  C0 0030B          ADDL2   #4, INTLIN                             ;
                                     15  11 00312          BRB     44$                                    ;
                      60  00000000'  EF  B0 00314 43$:     MOVW    P.AAN, (R0)                            ;
              00000000G          EF  02  C0 0031B          ADDL2   #2, LP                                 ;
              00000000G          EF  02  C0 00322          ADDL2   #2, INTLIN                             ;
                      57             01  D0 00329 44$:     MOVL    #1, DOING_UND                          ; 1117
                                     58  D4 0032C 45$:     CLRL    UND_CHAR                               ; 1120
                      20             56  D1 0032E 46$:     CMPL    CH, #32                                ; 1126
                                     0C  12 00331          BNEQ    48$                                    ;
                                     01  DD 00333          PUSHL   #1                                     ; 1127
              00000000V          EF  01  FB 00335          CALLS   #1, ENDWRD                             ;
                                   FD4C  31 0033C 47$:     BRW     9$                                     ;
                      3E             56  D1 0033F 48$:     CMPL    CH, #62                                ; 1129
```

```
                                                                L 14
FORMAT              FORMAT - generate formatted output lines   16-Sep-1984 00:34:26   VAX-11 Bliss-32 V4.0-742       Page  24
V04-000             FMTTXT - scan and format text              14-Sep-1984 13:06:27   [RUNOFF.SRC]FORMAT.BLI;1             (3)

                             03  13 00342        BEQL    49$                         ;
                           010B  31 00344        BRW     54$                         ;
             50 00000000G    EF  D0 00347  49$:  MOVL    LP, R0                      ; 1135
             03 00000000G    EF  D1 0034E        CMPL    MAJOR, #3                   ; 1132
                             03  13 00355        BEQL    50$                         ;
                           00E0  31 00357        BRW     52$                         ;
             60              56  90 0035A  50$:  MOVB    CH, (R0)                    ; 1135
                00000000G    EF  D6 0035D        INCL    LP                          ;
                00000000G    EF  D6 00363        INCL    INTLIN                      ;
                00000000G    EF  D6 00369        INCL    EXTLIN                      ;
C5 00000000G EF              01  E1 0036F        BBC     #1, CMDBLK, 47$             ; 1137
          10 AE 00000000G    EF  B0 00377        MOVW    INTLIN, $STR$STRING         ; 1143
          12 AE              0E  90 0037F        MOVB    #14, $STR$STRING+2          ;
          13 AE              01  90 00383        MOVB    #1, $STR$STRING+3           ;
          14 AE 00000000G    EF  9E 00387        MOVAB   LINE, $STR$STRING+4         ;
                00000000G    EF  9F 0038F        PUSHAB  STR$FAILURE                 ;
                             7E  D4 00395        CLRL    -(SP)                       ;
                00000000G    EF  9F 00397        PUSHAB  $STR$TARGET                 ;
                       1C    AE  9F 0039D        PUSHAB  $STR$STRING                 ;
                             7E  D4 003A0        CLRL    -(SP)                       ;
   00000000G EF              05  FB 003A2        CALLS   #5, XST$COPY                ;
             50 00000000G    EF  3C 003A9        MOVZWL  TMPSTR, R0                  ;
   00000000G EF              50  C0 003B0        ADDL2   R0, CHROUT                  ;
   00000000G EF 00000000G    EF  9E 003B7        MOVAB   $IOB$OUTPUT, IOB$+68        ;
   00000000G EF              07  90 003C2        MOVB    #7, IOB$+44                 ;
                00000000G    EF  9F 003C9        PUSHAB  XPO$FAILURE                 ;
                             7E  D4 003CF        CLRL    -(SP)                       ;
                00000000G    EF  9F 003D1        PUSHAB  IOB$                        ;
   00000000G EF              03  FB 003D7        CALLS   #3, XPO$PUT                 ;
07 00000000G EF              01  E1 003DE        BBC     #1, CMDBLK, 51$             ;
   00000000V EF              00  FB 003E6        CALLS   #0, SPLIT                   ;
   00000000G EF 00000000G    EF  9E 003ED  51$:  MOVAB   LINE, LP                    ; 1144
                00000000G    EF  D4 003F8        CLRL    INTLIN                      ;
                00000000G    EF  D4 003FE        CLRL    EXTLIN                      ;
   00000000G FF              3C  90 00404        MOVB    #60, @LP                    ; 1145
                00000000G    EF  D6 0040B        INCL    LP                          ;
                00000000G    EF  D6 00411        INCL    INTLIN                      ;
   00000000' EF 00000000G    EF  D0 00417        MOVL    LP, WRDPTR                  ; 1146
   00000000' EF 00000000G    EF  D0 00422        MOVL    EXTLIN, EXTWRD              ; 1147
   00000000' EF 00000000G    EF  D0 0042D        MOVL    INTLIN, INTWRD              ; 1148
                             59  11 00438        BRB     57$                         ; 1132
0A 00000000G EF              01  E1 0043A  52$:  BBC     #1, CMDBLK, 53$             ; 1155
             60 00000000'    EF  B0 00442        MOVW    P.AA0, (R0)                 ; 1157
                           01E4  31 00449        BRW     85$                         ;
             60              56  90 0044C  53$:  MOVB    CH, (R0)                    ; 1159
                           023D  31 0044F        BRW     89$                         ;
                             51  D4 00452  54$:  CLRL    R1                          ; 1165
             20              56  D1 00454        CMPL    CH, #32                     ;
                             02  18 00457        BGEQ    55$                         ;
                             51  D6 00459        INCL    R1                          ;
                             50  D4 0045B  55$:  CLRL    R0                          ;
   0000007E  8F              56  D1 0045D        CMPL    CH, #126                    ;
                             02  15 00464        BLEQ    56$                         ;
                             50  D6 00466        INCL    R0                          ;
             50              51  C8 00468  56$:  BISL2   R1, R0                      ;
             01              50  D1 0046B        CMPL    R0, #1                      ;
                             59  12 0046E        BNEQ    59$                         ;
```

```
                     1D 00000000G  EF           01  E1 00470        BBC     #1, CMDBLK, 58$                  ; 1171
                                   7E      04   AC  7D 00478        MOVQ    TXT_LEN, -(SP)                   ; 1179
                                                02  DD 0047C        PUSHL   #2                               ;
                                   00000000G    8F  DD 0047E        PUSHL   #USRTOC$_TEXTD                   ;
                                                7E  D4 00484        CLRL    -(SP)                            ;
                                   00000000G    8F  DD 00486        PUSHL   #DSRTOC$_CTRLCHAR                ;
                        00000000G  00           06  FB 0048C        CALLS   #6, LIB$SIGNAL                   ;
                                                6E  11 00493 57$:   BRB     62$                              ; 1171
                        00000000G  FF      5F   8F  90 00495 58$:   MOVB    #95, @LP                         ; 1192
                                   00000000G    EF  D6 0049D        INCL    LP                               ;
                                   00000000G    EF  D6 004A3        INCL    INTLIN                           ;
                        00000000G  FF           56  90 004A9        MOVB    CH, @LP                          ; 1193
                                   00000000G    EF  D6 004B0        INCL    LP                               ;
                                   00000000G    EF  D6 004B6        INCL    INTLIN                           ;
                                   08           56  D1 004BC        CMPL    CH, #8                           ; 1195
                                                42  12 004BF        BNEQ    62$                              ;
                                   00000000G    EF  D7 004C1        DECL    EXTLIN                           ; 1197
                                                3A  11 004C7        BRB     62$                              ; 1123
                     03 00000000G  EF           D1  004C9 59$:      CMPL    MAJOR, #3                        ; 1209
                                                03  12 004D0        BNEQ    60$                              ;
                                              01B3  31 004D2        BRW     88$                              ;
                     03 00000000G  EF           01  E0 004D5 60$:   BBS     #1, CMDBLK, 61$                  ; 1223
                                              0160  31 004DD        BRW     86$                              ;
                        0000005F   8F           56  D1 004E0 61$:   CMPL    CH, #95                          ; 1233
                                                1D  12 004E7        BNEQ    63$                              ;
           00000000G FF 00000000'  EF           05  28 004E9        MOVC3   #5, P.AAP, @LP                   ; 1234
                        00000000G  EF           05  C0 004F5        ADDL2   #5, LP                           ;
                        00000000G  EF           05  C0 004FC        ADDL2   #5, INTLIN                       ;
                                              FB85  31 00503 62$:   BRW     9$                               ; 1230
                                   2D           56  D1 00506 63$:   CMPL    CH, #45                          ; 1236
                                                0D  12 00509        BNEQ    64$                              ;
                        00000000G  FF 00000000' EF  B0 0050B        MOVW    P.AAQ, @LP                       ; 1237
                                                72  11 00516        BRB     70$                              ;
                                   2A           56  D1 00518 64$:   CMPL    CH, #42                          ; 1239
                                                0D  12 0051B        BNEQ    65$                              ;
                        00000000G  FF 00000000' EF  B0 0051D        MOVW    P.AAR, @LP                       ; 1240
                                                76  11 00528        BRB     72$                              ;
                                   3D           56  D1 0052A 65$:   CMPL    CH, #61                          ; 1242
                                                0D  12 0052D        BNEQ    66$                              ;
                        00000000G  FF 00000000' EF  B0 0052F        MOVW    P.AAS, @LP                       ; 1243
                                                7A  11 0053A        BRB     74$                              ;
                                   2B           56  D1 0053C 66$:   CMPL    CH, #43                          ; 1245
                                                0D  12 0053F        BNEQ    67$                              ;
                        00000000G  FF 00000000' EF  B0 00541        MOVW    P.AAT, @LP                       ; 1246
                                                7E  11 0054C        BRB     76$                              ;
                        0000005C   8F           56  D1 0054E 67$:   CMPL    CH, #92                          ; 1248
                                                0D  12 00555        BNEQ    68$                              ;
                        00000000G  FF 00000000' EF  B0 00557        MOVW    P.AAU, @LP                       ; 1249
                                                7A  11 00562        BRB     78$                              ;
                        00000040   8F           56  D1 00564 68$:   CMPL    CH, #64                          ; 1251
                                                0D  12 0056B        BNEQ    69$                              ;
                        00000000G  FF 00000000' EF  B0 0056D        MOVW    P.AAV, @LP                       ; 1252
                                                7A  11 00578        BRB     80$                              ;
                                   2F           56  D1 0057A 69$:   CMPL    CH, #47                          ; 1254
                                                0D  12 0057D        BNEQ    71$                              ;
                        00000000G  FF 00000000' EF  B0 0057F        MOVW    P.AAW, @LP                       ; 1255
                                                7E  11 0058A 70$:   BRB     82$                              ;
                     0000007C   8F                  56  D1 0058C 71$:    CMPL    CH, #124                          ; 1257
                                                    0D  12 00593         BNEQ    73$
                     00000000G  FF 00000000'        EF  B0 00595         MOVW    P.AAX, @LP                        ; 1258
                                                    68  11 005A0 72$:    BRB     82$
                     0000007B   8F                  56  D1 005A2 73$:    CMPL    CH, #123                          ; 1260
                                                    0D  12 005A9         BNEQ    75$
                     00000000G  FF 00000000'        EF  B0 005AB         MOVW    P.AAY, @LP                        ; 1261
                                                    78  11 005B6 74$:    BRB     85$
                     0000007D   8F                  56  D1 005B8 75$:    CMPL    CH, #125                          ; 1263
                                                    0D  12 005BF         BNEQ    77$
                     00000000G  FF 00000000'        EF  B0 005C1         MOVW    P.AAZ, @LP                        ; 1264
                                                    62  11 005CC 76$:    BRB     85$
                                3C                  56  D1 005CE 77$:    CMPL    CH, #60                           ; 1266
                                                    0D  12 005D1         BNEQ    79$
                     00000000G  FF 00000000'        EF  B0 005D3         MOVW    P.ABA, @LP                        ; 1267
                                                    50  11 005DE 78$:    BRB     85$
                     0000005B   8F                  56  D1 005E0 79$:    CMPL    CH, #91                           ; 1269
                                                    0D  12 005E7         BNEQ    81$
                     00000000G  FF 00000000'        EF  B0 005E9         MOVW    P.ABB, @LP                        ; 1270
                                                    3A  11 005F4 80$:    BRB     85$
                     0000005D   8F                  56  D1 005F6 81$:    CMPL    CH, #93                           ; 1272
                                                    0D  12 005FD         BNEQ    83$
                     00000000G  FF 00000000'        EF  B0 005FF         MOVW    P.ABC, @LP                        ; 1273
                                                    24  11 0060A 82$:    BRB     85$
                                22                  56  D1 0060C 83$:    CMPL    CH, #34                           ; 1275
                                                    77  12 0060F         BNEQ    88$
                                50 00000000G        EF  D0 00611         MOVL    LP, R0                            ; 1284
                                0B                  59  E9 00618         BLBC    OPEN_QUOTE, 84$                   ; 1278
                                60 00000000'        EF  B0 0061B         MOVW    P.ABD, (R0)                       ; 1284
                                                    59  D4 00622         CLRL    OPEN_QUOTE                        ; 1285
                                                    0A  11 00624         BRB     85$                               ; 1278
                                60 00000000'        EF  B0 00626 84$:    MOVW    P.ABE, (R0)                       ; 1292
                                59                  01  D0 0062D         MOVL    #1, OPEN_QUOTE                    ; 1293
                     00000000G  EF                  02  C0 00630 85$:    ADDL2   #2, LP                            ; 1284
                     00000000G  EF                  02  C0 00637         ADDL2   #2, INTLIN
                                                    61  11 0063E         BRB     90$                               ; 1230
                     0000005F   8F                  56  D1 00640 86$:    CMPL    CH, #95                           ; 1312
                                                    28  13 00647         BEQL    87$
                                2A                  56  D1 00649         CMPL    CH, #42                           ; 1313
                                                    26  13 0064C         BEQL    87$
                                21                  56  D1 0064E         CMPL    CH, #33                           ; 1314
                                                    21  13 00651         BEQL    87$
                                2E                  56  D1 00653         CMPL    CH, #46                           ; 1315
                                                    1C  13 00656         BEQL    87$
                     0000005C   8F                  56  D1 00658         CMPL    CH, #92                           ; 1316
                                                    13  13 0065F         BEQL    87$
                                25                  56  D1 00661         CMPL    CH, #37                           ; 1317
                                                    0E  13 00664         BEQL    87$
                                26                  56  D1 00666         CMPL    CH, #38                           ; 1318
                                                    09  13 00669         BEQL    87$
                     0000005E   8F                  56  D1 0066B         CMPL    CH, #94                           ; 1319
                                                    14  12 00672         BNEQ    88$
                     00000000G  FF          5F      8F  90 00674 87$:    MOVB    #95, @LP                          ; 1324
                                   00000000G        EF  D6 0067C         INCL    LP
                                   00000000G        EF  D6 00682         INCL    INTLIN
                     00000000G  FF                  56  90 00688 88$:    MOVB    CH, @LP                           ; 1326
                                   00000000G        EF  D6 0068F 89$:    INCL    LP
```

```
                                    00000000G  EF      D6 00695            INCL    INTLIN                                 ;
                                    00000000G  EF      D6 0069B            INCL    EXTLIN                                 ;
                                             F9E7      31 006A1  90$:      BRW     5$                                     ; 1123
                                                7E     D4 006A4  91$:      CLRL    -(SP)                                  ; 1338
                           00000000V  EF               01      FB 006A6            CALLS   #1, ENDWRD                             ;
                        75 00000000G  EF               01      E1 006AD            BBC     #1, CMDBLK, 95$                        ; 1340
                                      03       08      AE      E8 006B5            BLBS    DOING_BOLD, 92$                        ; 1344
                                      4F               57      E9 006B9            BLBC    DOING_UND, 94$                         ;
00000000G  FF           18            00 00000000'     EF      F0 006BC  92$:      INSV    P.ABF, #0, #24, @LP                    ; 1350
                           00000000G  EF               03      C0 006C9            ADDL2   #3, LP                                 ;
                           00000000G  EF               03      C0 006D0            ADDL2   #3, INTLIN                             ;
                                      1C       08      AE      E9 006D7            BLBC    DOING_BOLD, 93$                        ; 1355
                                      19               57      E9 006DB            BLBC    DOING_UND, 93$                         ;
                           00000000G  FF 00000000'     EF      B0 006DE            MOVW    P.ABG, @LP                             ;
                           00000000G  EF               02      C0 006E9            ADDL2   #2, LP                                 ;
                           00000000G  EF               02      C0 006F0            ADDL2   #2, INTLIN                             ;
                           00000000G  FF       5D      8F      90 006F7  93$:      MOVB    #93, @LP                               ; 1357
                                    00000000G  EF      D6 006FF            INCL    LP                                     ;
                                    00000000G  EF      D6 00705            INCL    INTLIN                                 ;
                                      55               59      E8 0070B  94$:      BLBS    OPEN_QUOTE, 97$                        ; 1360
                                      7E       04      AC      7D 0070E            MOVQ    TXT_LEN, -(SP)                         ; 1367
                                                       02      DD 00712            PUSHL   #2                                     ;
                                    00000000G          8F      DD 00714            PUSHL   #DSRTOC$_TEXTD                         ;
                                                       7E      D4 0071A            CLRL    -(SP)                                  ;
                                    00000000G          8F      DD 0071C            PUSHL   #DSRTOC$_CLOSEQUOT                     ;
                           00000000G  00               06      FB 00722            CALLS   #6, LIB$SIGNAL                         ;
                                                       04 00729            RET                                    ; 1340
                                      19       08      AE      E9 0072A  95$:      BLBC    DOING_BOLD, 96$                        ; 1381
                           00000000G  FF 00000000'     EF      B0 0072E            MOVW    P.ABH, @LP                             ;
                           00000000G  EF               02      C0 00739            ADDL2   #2, LP                                 ;
                           00000000G  EF               02      C0 00740            ADDL2   #2, INTLIN                             ;
                                      19               57      E9 00747  96$:      BLBC    DOING_UND, 97$                         ; 1383
                           00000000G  FF 00000000'     EF      B0 0074A            MOVW    P.ABI, @LP                             ;
                           00000000G  EF               02      C0 00755            ADDL2   #2, LP                                 ;
                           00000000G  EF               02      C0 0075C            ADDL2   #2, INTLIN                             ;
                                                       04 00763  97$:      RET                                    ; 1387

; Routine Size:  1892 bytes,    Routine Base:  $CODE$ + 00A3
```

```
%SBTTL 'ENDWRD - verify word fits on line'
ROUTINE endwrd (space) : NOVALUE =
!++
! FUNCTIONAL DESCRIPTION:
!
!       ENDWRD is called when a space is about to be output. For RUNOFF,
!       it makes sure that the word that the space ends fits on the the line.
!       If it doesn't, it wraps the line.
!
! FORMAL PARAMETERS:
!
!       space           - true if a space is to be generated
!
! IMPLICIT INPUTS:
!
!       cmdblk          - command line information block
!       rmargin         - indicates how far to the right this word can extend.
!       line_indent     - number of columns of indentation
!       wrap            - first column to start new line in
!       wrdptr          - pointer to beginning of word
!       extwrd          - external length of line not including word
!       intwrd          - internal length of line not including word
!
! IMPLICIT OUTPUTS:
!
!       wrdptr          - points to the end of the current word
!       extwrd          - new external line length
!       intwrd          - new internal line length
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    IF (.extlin GTR .rmargin) AND ( NOT .cmdblk [contents$v_tms11])
    THEN
        BEGIN
        !
        ! The word that this space terminates does not fit.  Wrap the line.
        ! First determine the length of the word just ended.
        ! Note that WORD_xxxxx were set at the beginning of the word being
        ! terminated, while the normal counters have been updated ever since.
        !
        extwrd = .extlin - .extwrd;
        intwrd = .intlin - .intwrd;
        !
        ! Now adjust the current line lengths before outputting the line
        !
        extlin = .extlin - .extwrd;
```

```
3           intlin = .intlin - .intwrd;
3           !
3           ! Before outputting the line that is to be wrapped, make sure that
3           ! at least two lines are still available on the page.  This avoids
3           ! having the first part of the text on one page and the last part
3           ! of it on another page.
3           !
3           put ('.TEST PAGE 2');
3           !
3           ! And now output the line, up to but not including the word that
3           ! this space terminates.
3           !
3           put ((.intlin, CH$PTR (line)));
3           clr_line ();
3           !
3           ! Add sufficient spaces to align the wrapped word with the first
3           ! character of the line that was just terminated.
3           !
3           pad ((.wrap - .line_indent));
3           !
3           ! Adjust the external line length.
3           ! It really represents .line_indent additional characters.
3           !
3           extlin = .extlin + .line_indent;
3           !
3           ! At this point the word that would have overflowed the line is
3           ! sitting out in limbo. But, we know its length and where it is.
3           ! Move it to the left so that it's aligned properly.
3           !
3
3           INCR i FROM 1 TO .intwrd DO
3               CH$WCHAR_A (CH$RCHAR_A (wrdptr), lp);
3
3           !
3           ! And finally, update the counters that were bypassed in the move
3           !
3           extlin = .extlin + .extwrd;
3           intlin = .intlin + .intwrd;
3           END;
2
2       IF .space THEN write_char (%C' ', counts_visually);
2
2       !
2       ! Remember current lengths for use the next time around.
2       !
2       extwrd = .extlin;
2       intwrd = .intlin;
2       wrdptr = .lp;
1       END;                                        ! End of endwrd


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

32 20 45 47 41 50 20 54 53 45 54 2E  00090 P.ABJ:  .ASCII  \.TEST PAGE 2\                                  ;

                                                            .PSECT  $OWN$,NOEXE,2
```

```
                                                  000C  00018 $STR$STRING:
                                                                        .WORD    12                                             :
                                              01  0E  0001A             .BYTE    14, 1                                          :
                                            00000000' 0001C             .ADDRESS P.ABJ                                          :



                                                                        .PSECT   $CODE$,NOWRT,2

                                                  0FFC 00000 ENDWRD:    .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1389
                       5B 00000000G   EF     9E 00002             MOVAB    LP, R11                                        :
                       5A 00000000G   EF     9E 00009             MOVAB    INTLIN, R10                                    :
                       59 00000000G   EF     9E 00010             MOVAB    EXTLIN, R9                                     :
                       58 00000000'   EF     9E 00017             MOVAB    INTWRD, R8                                     :
                       5E             08     C2 0001E             SUBL2    #8, SP                                         :
          00000000G    EF             69     D1 00021             CMPL     EXTLIN, RMARGIN                                ; 1430
                                      03     14 00028             BGTR     2$                                             :
                                    0137     31 0002A 1$:         BRW      8$                                             :
     F5   00000000G    EF             01     E0 0002D 2$:         BBS      #1, CMDBLK, 1$                                 :
  FC  A8               69       FC    A8     C3 00035             SUBL3    EXTWRD, EXTLIN, EXTWRD                         ; 1439
      68               6A             68     C3 0003B             SUBL3    INTWRD, INTLIN, INTWRD                         ; 1440
                       69       FC    A8     C2 0003F             SUBL2    EXTWRD, EXTLIN                                 ; 1444
                       6A             68     C2 00043             SUBL2    INTWRD, INTLIN                                 ; 1445
                          00000000G   EF     9F 00046             PUSHAB   STR$FAILURE                                    ; 1452
                                      7E     D4 0004C             CLRL     -(SP)                                          :
                          00000000G   EF     9F 0004E             PUSHAB   $STR$TARGET                                    :
                                04    A8     9F 00054             PUSHAB   $STR$STRING                                    :
                                      7E     D4 00057             CLRL     -(SP)                                          :
          00000000G    EF             05     FB 00059             CALLS    #5, XST$COPY                                   :
                       50 00000000G   EF     3C 00060             MOVZWL   TMPSTR, R0                                     :
          00000000G    EF             50     C0 00067             ADDL2    R0, CHROUT                                     :
          00000000G    EF 00000000G   EF     9E 0006E             MOVAB    $IOB$OUTPUT, IOB$+68                           :
          00000000G    EF             07     90 00079             MOVB     #7, IOB$+44                                    :
                          00000000G   EF     9F 00080             PUSHAB   XPO$FAILURE                                    :
                                      7E     D4 00086             CLRL     -(SP)                                          :
                          00000000G   EF     9F 00088             PUSHAB   IOB$                                           :
          00000000G    EF             03     FB 0008E             CALLS    #3, XPO$PUT                                    :
       07 00000000G    EF             01     E1 00095             BBC      #1, CMDBLK, 3$                                 :
          00000000V    EF             00     FB 0009D             CALLS    #0, SPLIT                                      :
                       6E             6A     B0 000A4 3$:         MOVW     INTLIN, $STR$STRING                            ; 1457
                  02   AE             0E     90 000A7             MOVB     #14, $STR$STRING+2                             :
                  03   AE             01     90 000AB             MOVB     #1, $STR$STRING+3                              :
                  04   AE 00000000G   EF     9E 000AF             MOVAB    LINE, $STR$STRING+4                            :
                          00000000G   EF     9F 000B7             PUSHAB   STR$FAILURE                                    :
                                      7E     D4 000BD             CLRL     -(SP)                                          :
                          00000000G   EF     9F 000BF             PUSHAB   $STR$TARGET                                    :
                                0C    AE     9F 000C5             PUSHAB   $STR$STRING                                    :
                                      7E     D4 000C8             CLRL     -(SP)                                          :
          00000000G    EF             05     FB 000CA             CALLS    #5, XST$COPY                                   :
                       50 00000000G   EF     3C 000D1             MOVZWL   TMPSTR, R0                                     :
          00000000G    EF             50     C0 000D8             ADDL2    R0, CHROUT                                     :
          00000000G    EF 00000000G   EF     9E 000DF             MOVAB    $IOB$OUTPUT, IOB$+68                           :
          00000000G    EF             07     90 000EA             MOVB     #7, IOB$+44                                    :
                          00000000G   EF     9F 000F1             PUSHAB   XPO$FAILURE                                    :
                                      7E     D4 000F7             CLRL     -(SP)                                          :
                          00000000G   EF     9F 000F9             PUSHAB   IOB$                                           :
```

```
                       00000000G  EF              03  FB 000FF        CALLS   #3, XPO$PUT                             ;
                    07 00000000G  EF              01  E1 00106        BBC     #1, CMDBLK, 4$                          ;
                       00000000V  EF              00  FB 0010E        CALLS   #0, SPLIT                               ;
                              6B 00000000G  EF    9E 00115 4$:        MOVAB   LINE, LP                                ; 1458
                                            6A    D4 0011C            CLRL    INTLIN                                  ;
                                            69    D4 0011E            CLRL    EXTLIN                                  ;
                              57 00000000G  EF    D0 00120            MOVL    LINE_INDENT, R7                         ; 1463
                              57 00000000G  EF    D1 00127            CMPL    WRAP, R7                                ;
                                            18    15 0012E            BLEQ    5$                                      ;
                    56 00000000G  EF              57  C3 00130        SUBL3   R7, WRAP, R6                            ;
          56        20            6E              00  2C 00138        MOVC5   #0, (SP), #32, R6, @LP                  ;
                                       00         BB     0013D                                                        ;
                              6B                  56  C0 0013F        ADDL2   R6, LP                                  ;
                              6A                  56  C0 00142        ADDL2   R6, INTLIN                              ;
                              69                  56  C0 00145        ADDL2   R6, EXTLIN                              ;
                              69                  57  C0 00148 5$:    ADDL2   R7, EXTLIN                              ; 1468
                                                  50  D4 0014B        CLRL    I                                       ; 1476
                                                  0A  11 0014D        BRB     7$                                      ;
                         00   BB            F8    B8  90 0014F 6$:    MOVB    @WRDPTR, @LP                            ;
                                            F8    A8  D6 00154        INCL    WRDPTR                                  ;
                                                  6B  D6 00157        INCL    LP                                      ;
                    F2        50                  68  F3 00159 7$:    AOBLEQ  INTWRD, I, 6$                           ;
                              69            FC    A8  C0 0015D        ADDL2   EXTWRD, EXTLIN                          ; 1481
                              6A                  68  C0 00161        ADDL2   INTWRD, INTLIN                          ; 1482
                              0A            04    AC  E9 00164 8$:    BLBC    SPACE, 9$                               ; 1485
                         00   BB                  20  90 00168        MOVB    #32, @LP                                ;
                                                  6B  D6 0016C        INCL    LP                                      ;
                                                  6A  D6 0016E        INCL    INTLIN                                  ;
                                                  69  D6 00170        INCL    EXTLIN                                  ;
                         FC   A8                  69  D0 00172 9$:    MOVL    EXTLIN, EXTWRD                          ; 1490
                              68                  6A  D0 00176        MOVL    INTLIN, INTWRD                          ; 1491
                         F8   A8                  6B  D0 00179        MOVL    LP, WRDPTR                              ; 1492
                                                      04 0017D        RET                                             ; 1493

; Routine Size:  382 bytes,    Routine Base:  $CODE$ + 0807
```

```
1 %SBTTL 'SPLIT - start new output file for tms if necessary'
1 GLOBAL ROUTINE split : NOVALUE =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine checks to see if the TMS output must be split
1 !       to another output file. This is necessary to prevent long
1 !       galleys which could jam the typesetter.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       None
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       chrout - number of characters written to the current output file.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
1
2     BEGIN
2
2     IF .chrout GEQ tms_characters_per_file
2     THEN
3         BEGIN
3         !
3         ! Must start a new output file
3         !
3
3         LOCAL
3             name_len,
3             spec_blk : $xpo_spec_block;
3
3         IF .outfile [str$h_length] EQL 0
3         THEN
3             !
3             ! Save current output file name
3             !
3             $str_copy (string = tocoob [iob$t_resultant], target = outfile);
3
3         !
3         ! Write terminator to current file and close it
3         !
3         $xpo_put (iob = tocoob, string = '*cfini*');
3         $xpo_close (iob = tocoob);
```

```
      3             !
      3             ! Compute new file name
      3             !
      3             $xpo_parse_spec (file_spec = outfile, spec_block = spec_blk);
      3             name_len = (IF .spec_blk [xpo$h_file_name] GEQ 6 THEN 6 ELSE .spec_blk [xpo$h_file_name]);
      3             fileno = .fileno + 1;
      3             !
      3             ! Initialize IOB, open new file and reset character count
      3             !
      3             $xpo_iob_init (iob = tocoob);
P     3             $xpo_open (iob = tocoob, options = output, default = outfile,
P     3                 file_spec = $str_concat ((.name_len, .spec_blk [xpo$a_file_name]),
P     3                     $str_ascii (.fileno, UNSIGNED, leading_zero, length = 3))
P     3                         %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
      3             );
      3             chrout = 0;
      3
      3             !
      3             ! Tell user about new file
      3             !
L     3 %IF %BLISS (BLISS32)
      3 %THEN
      3             SIGNAL (contents$_tms11, 1, tocoob [iob$t_resultant]);
U     3 %ELSE
U     3             $xpo_put_msg (severity = success,
U     3                 string = $str_concat ('output file full - continuing with file ''',
U     3                     tocoob [iob$t_resultant], ''''));
      3 %FI
      3
      3             !
      3             ! Write file prologue
      3             !
      3             put ('*start*');
      3             put ('*cinit*');
      2             END;
      2
      1         END;



                                                            .PSECT  $SPLIT$,NOWRT,NOEXE,2

            2A  69  6E  69  66  63  2A  0009C P.ABK:  .ASCII  \*cfini*\                     ;
            2A  74  72  61  74  73  2A  000A3 P.ABL:  .ASCII  \*start*\                     ;
            2A  74  69  6E  69  63  2A  000AA P.ABM:  .ASCII  \*cinit*\                     ;

                                                            .PSECT  $OWN$,NOEXE,2

                                  0007  00020 $IOB$OUTPUT:
                                                            .WORD   7                       ;
                              01  0E  00022                 .BYTE   14, 1                   ;
                           00000000'  00024                 .ADDRESS P.ABK                  ;
                                  0007  00028 $STR$STRING:
                                                            .WORD   7                       ;
                              01  0E  0002A                 .BYTE   14, 1                   ;
                           00000000'  0002C                 .ADDRESS P.ABL                  ;
                                  0007  00030 $STR$STRING:
```

```
                                                                     .WORD    7                                               ;
                                               01  0E  00032         .BYTE    14, 1                                           ;
                                            00000000'  00034         .ADDRESS P.ABM                                           ;

                                                             $STR$TARGET=       OUTFILE
                                                             $STR$FILE_SPEC=    OUTFILE
                                                             $IOB$DEFAULT=      OUTFILE
                                                                     .EXTRN   XPO$CLOSE, XPO$PARSE_SPEC
                                                                     .EXTRN   XST$JOIN, XST$ASCII
                                                                     .EXTRN   XPO$OPEN

                                                                     .PSECT   $CODE$,NOWRT,2

                                             0FFC  00000             .ENTRY   SPLIT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 ; 1495
                        5B 00000000G  EF  9E  00002             MOVAB    CHROUT, R11                                     ;
                        5A 00000000G  EF  9E  00009             MOVAB    XPO$FAILURE, R10                                ;
                        59 00000000G  EF  9E  00010             MOVAB    $STR$TARGET, R9                                 ;
                        58 00000000'  EF  9E  00017             MOVAB    OUTFILE, R8                                     ;
                        57 00000000G  EF  9E  0001E             MOVAB    IOB$, R7                                        ;
                        5E        B0  AE  9E  00025             MOVAB    -80(SP), SP                                     ;
           00002800  8F               6B  D1  00029             CMPL     CHROUT, #10240                                  ; 1528
                                      01  18  00030             BGEQ     1$                                              ;
                                          04  00032             RET                                                      ;
                                      68  B5  00033  1$:        TSTW     OUTFILE                                         ; 1539
                                      16  12  00035             BNEQ     2$                                              ;
                          00000000G   EF  9F  00037             PUSHAB   STR$FAILURE                                     ; 1544
                                      7E  D4  0003D             CLRL     -(SP)                                           ;
                                      58  DD  0003F             PUSHL    R8                                              ;
                                1C    A7  9F  00041             PUSHAB   $STR$STRING                                     ;
                                      7E  D4  00044             CLRL     -(SP)                                           ;
           00000000G  EF              05  FB  00046             CALLS    #5, XST$COPY                                    ;
                  44  A7        1C    A8  9E  0004D  2$:        MOVAB    $IOB$OUTPUT, IOB$+68                            ; 1549
                  2C  A7              07  90  00052             MOVB     #7, IOB$+44                                     ;
                                      5A  DD  00056             PUSHL    R10                                             ;
                                      7E  D4  00058             CLRL     -(SP)                                           ;
                                      57  DD  0005A             PUSHL    R7                                              ;
           00000000G  EF              03  FB  0005C             CALLS    #3, XPO$PUT                                     ;
                  2C  A7              02  90  00063             MOVB     #2, IOB$+44                                     ; 1550
                                      5A  DD  00067             PUSHL    R10                                             ;
                                      7E  D4  00069             CLRL     -(SP)                                           ;
                                      57  DD  0006B             PUSHL    R7                                              ;
           00000000G  EF              03  FB  0006D             CALLS    #3, XPO$CLOSE                                   ;
                                      5A  DD  00074             PUSHL    R10                                             ; 1554
                                      7E  D4  00076             CLRL     -(SP)                                           ;
                      7E              01  CE  00078             MNEGL    #1, -(SP)                                       ;
                                14    AE  9F  0007B             PUSHAB   SPEC_BLK                                        ;
                                      58  DD  0007E             PUSHL    R8                                              ;
           00000000G  EF              05  FB  00080             CALLS    #5, XPO$PARSE_SPEC                              ;
                      06        28    AE  B1  00087             CMPW     SPEC_BLK+32, #6                                 ; 1555
                                      05  1F  0008B             BLSSU    3$                                              ;
                      56              06  D0  0008D             MOVL     #6, NAME_LEN                                    ;
                                      04  11  00090             BRB      4$                                              ;
                      56        28    AE  3C  00092  3$:        MOVZWL   SPEC_BLK+32, NAME_LEN                           ;
                                FC    A8  D6  00096  4$:        INCL     FILENO                                          ; 1556
   00F4   8F        00        6E              00  2C  00099             MOVC5    #0, (SP), #0, #244, IOB$                        ; 1560
                                      67      000A0                                                                      ;
                      67 0301003D     8F  D0  000A1             MOVL     #50397245, IOB$                                 ;
           1E    A7       020E     8F  B0 000A8         MOVW     #526, IOB$RESULTANT+2                  ;
                                   03  DD 000AE         PUSHL    #3                                     ; 1565
                            FC     A8  DD 000B0         PUSHL    FILENO                                 ;
                 7E       0603     8F  3C 000B3         MOVZWL   #1539, -(SP)                           ;
    00000000G    EF                03  FB 000B8         CALLS    #3, XST$ASCII                          ;
                 6E                56  B0 000BF         MOVW     NAME_LEN, $STR$STRING0                 ;
           02    AE                0E  90 000C2         MOVB     #14, $STR$STRING0+2                    ;
           03    AE                01  90 000C6         MOVB     #1, $STR$STRING0+3                     ;
           04    AE         2C     AE  D0 000CA         MOVL     SPEC_BLK+36, $STR$STRING0+4            ;
                                   50  DD 000CF         PUSHL    R0                                     ;
                            04     AE  9F 000D1         PUSHAB   $STR$STRING0                           ;
    00000000G    EF                02  FB 000D4         CALLS    #2, XST$JOIN                           ;
           04    A7                50  D0 000DB         MOVL     R0, IOB$+4                             ;
           08    A7                68  9E 000DF         MOVAB    $IOB$DEFAULT, IOB$+8                   ;
           2E    A7                02  88 000E3         BISB2    #2, IOB$+46                            ;
           2C    A7                01  90 000E7         MOVB     #1, IOB$+44                            ;
                          00000000G  EF  9F 000EB       PUSHAB   OPEN_ERROR                             ;
                                   7E  D4 000F1         CLRL     -(SP)                                  ;
                                   57  DD 000F3         PUSHL    R7                                     ;
    00000000G    EF                03  FB 000F5         CALLS    #3, XPO$OPEN                           ;
                                   6B  D4 000FC         CLRL     CHROUT                                 ; 1566
                            1C     A7  9F 000FE         PUSHAB   TOCOOB+28                              ; 1573
                                   01  DD 00101         PUSHL    #1                                     ;
                          00000000G  8F  DD 00103       PUSHL    #DSRTOC$_TMS11                         ;
    00000000G    00                03  FB 00109         CALLS    #3, LIB$SIGNAL                         ;
                          00000000G  EF  9F 00110       PUSHAB   STR$FAILURE                            ; 1583
                                   7E  D4 00116         CLRL     -(SP)                                  ;
                                   59  DD 00118         PUSHL    R9                                     ;
                            24     A8  9F 0011A         PUSHAB   $STR$STRING                            ;
                                   7E  D4 0011D         CLRL     -(SP)                                  ;
    00000000G    EF                05  FB 0011F         CALLS    #5, XST$COPY                           ;
                 50                69  3C 00126         MOVZWL   TMPSTR, R0                             ;
                 6B                50  C0 00129         ADDL2    R0, CHROUT                             ;
           44    A7                69  9E 0012C         MOVAB    $IOB$OUTPUT, IOB$+68                   ;
           2C    A7                07  90 00130         MOVB     #7, IOB$+44                            ;
                                   5A  DD 00134         PUSHL    R10                                    ;
                                   7E  D4 00136         CLRL     -(SP)                                  ;
                                   57  DD 00138         PUSHL    R7                                     ;
    00000000G    EF                03  FB 0013A         CALLS    #3, XPO$PUT                            ;
 05 00000000G    EF                01  E1 00141         BBC      #1, CMDBLK, 5$                         ;
         FEB2    CF                00  FB 00149         CALLS    #0, SPLIT                              ;
                          00000000G  EF  9F 0014E  5$:  PUSHAB   STR$FAILURE                            ; 1584
                                   7E  D4 00154         CLRL     -(SP)                                  ;
                                   59  DD 00156         PUSHL    R9                                     ;
                            2C     A8  9F 00158         PUSHAB   $STR$STRING                            ;
                                   7E  D4 0015B         CLRL     -(SP)                                  ;
    00000000G    EF                05  FB 0015D         CALLS    #5, XST$COPY                           ;
                 50                69  3C 00164         MOVZWL   TMPSTR, R0                             ;
                 6B                50  C0 00167         ADDL2    R0, CHROUT                             ;
           44    A7                69  9E 0016A         MOVAB    $IOB$OUTPUT, IOB$+68                   ;
           2C    A7                07  90 0016E         MOVB     #7, IOB$+44                            ;
                                   5A  DD 00172         PUSHL    R10                                    ;
                                   7E  D4 00174         CLRL     -(SP)                                  ;
                                   57  DD 00176         PUSHL    R7                                     ;
    00000000G    EF                03  FB 00178         CALLS    #3, XPO$PUT                            ;
 05 00000000G    EF                01  E1 0017F         BBC      #1, CMDBLK, 6$                         ;
         FE74    CF                00  FB 00187         CALLS    #0, SPLIT                              ;
```

GCO
V04

; R

;

```
                                                   04 0018C 6$:      RET                                                      ; 1587

; Routine Size:  397 bytes,    Routine Base:  $CODE$ + 0985


; 1117          1588  1
; 1118          1589  1 END                                                   ! End of module
; 1119          1590  0 ELUDOM



                                                                        .EXTRN  LIB$SIGNAL
;                                   PSECT SUMMARY
;
;       Name                      Bytes                     Attributes
;
;  $OWN$                             56  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                          2834  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                           177  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                              Library Statistics
;
;                                                -------- Symbols --------      Pages       Processing
;       File                                     Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]XPORT.L32;1                 590       202        34        252         00:00.1




;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:FORMAT/OBJ=OBJ$:FORMAT MSRC$:FORMAT/UPDATE=(ENH$:FORMAT)

; Size:           2834 code + 233 data bytes
; Run Time:          01:07.9
; Elapsed Time:      02:31.0
; Lines/CPU Min:       1404
; Lexemes/CPU-Min: 61791
; Memory Used:  594 pages
; Compilation Complete
```

0341 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

FIND
LIS

ENDWRD
LIS

ERROR
LIS

FIGURE
LIS

FLGSEM
LIS

FOOFIL
LIS

GCODE
LIS

FCIMRA
LIS

FNONLY
LIS

FUNFNU
LIS

FOOBOT
LIS

GBLDCL
LIS

FNDFLG
LIS

FOOOUT
LIS

FORMAT
LIS